# CENG242 Homework 2 Graphical User Interface User Manual

## 1 Installation

Since Haskell does not have a huge standard library (unlike Java, for example), the GUI program I wrote for you relies heavily on third party libraries. Naturally, this means that the installation process will be a bit involved! Here are the steps you should try to follow (for Ubuntu):

1. Install pre-requisite packages:
   `sudo apt-get install build-essential autogen autoconf mesa-utils libxft-dev git`
2. Install the Haskell project managing tool, stack:
   `wget -qO- https://get.haskellstack.org/ | sh`
3. You may need to perform an update at this step:
   `stack upgrade`
4. Clone the git repository containing the code:
   `git clone https://github.com/denizmsayin/hw2-gui`
5. Build the project for the first time, this will take a while and require an internet connetion since some third-party libraries have to be installed:
   `cd hw2-gui && stack build --flag fltkhs:bundled`
6. Once this is complete, you can run the GUI as follows:
   `stack exec hw2-gui`
7. When you update your HW2 code (which is under `src/`), you should also rebuild the project:
   `stack build`

Send me an email if you get any errors during installation on Ubuntu, some necessary libraries might be missing. Unfortunately I cannot provide any support for other distributions, Mac or Windows.

Also note that you can find and modify the compiler flags for the build can be changed through the first `ghc-Options:` item in the `hw2-gui.cabal` file.

## 2 Features

The aim of the GUI is to ease the visualization of the trees so that you can focus on your program logic. Here's a description of what each part does:

- **Slot selection:** Select the slot that you are working with. Initially, the GUI only has a single active slot. Each slot can contain one expression tree and a title.
- **Text input:** This box is where you provide all your text input. It is possible to break your input over multiple lines.
- **Read:** Tries to read the text input as an expression by using the standard (deriving Read) definition of the expression structure. This is useful for copying trees from ghci's output into the GUI to visualize them.

- **Parse:** Tries to parse the text using the `parse` function from the parser module, e.g. `a+b+(-c*3)` is a valid input. Note that the parser ignores all whitespace.
- **Show:** Shows the current expression on stdout. This is useful to copy your expression somewhere else, such as a ghci instance.
- **Set Slot Title:** The current slot's title is set to the text in the input box. The title will appear over the expression in the slot.
- **Parse Assignments:** Parses assignments from the text input. Each assignment should be on a separate line: `a = 1 + 2 b= a+3 c = x * 7` All existing slots will be overwritten with the provided assignments from left to right.
- **Fold & Propagate Constants:** This applies the foldAndPropagateConstants function to the existing slots, using titles as variable names. Ideal for being used in conjunction with the 'Parse Assignments' button. This button is hooked to the definition provided in the HW2.hs file, so initially it contains a dummy implementation. Once you implement it properly, you can see your results right here!
- **Assign Common Subexprs:** Applies the assignCommonSubexprs function to the expression in the current slot, and replaces all slots with the resulting assignments, the reduced main expression being the rightmost. Just like the previous button, the result depends on your own definition in the HW2.hs file.
- **Reduce Polynomial:** Applies the reducePoly function to the current slot and replaces the expression at that slot with the result. Tied to your own definition in HW2.hs, just like the two previous buttons.
- **Drawing:** From this menu, you can save the current drawing as a PNG file, which might be useful for discussing with your friends or asking questions. You can also add and remove new slots, as well as open a configuration menu which contains lots of settings. Using the 'Persist' button in the configuration menu will save your settings to a file which will be reloaded on startup. Note that the height & width settings simply set the dimensions on startup, the window is resizeable.

I hope the GUI eases your implementation! I tried to handle most errors, but you can email me any bugs you find, using `sayin@ceng.metu.edu.tr`. Obviously, throwing errors from the functions you are supposed to define will crash the GUI.

Remember that installing this GUI is entirely optional and not related to your grading in any way. It's simply a tool for visualizing your results.